This document has not yet been approved by the
Government for general use or distribution

Managing Events
Functional Requirements
for the
Hubble Space Telescope
Control Center System

Prepared for
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

Prepared by
System Monitoring Team
of the
Vision 2000 CCS Project

December 1996

This document has not yet been approved by the

Government for general use or distribution

Abstract

This document provides the high level system functional requirements for the Managing Events (EVT) process within the Control Center System (CCS) of the Hubble Space Telescope (HST) under the control of the Vision 2000 project . The EVT within the System Monitoring (SYM) subsystem will generate, filter, route, and log all CCS event messages.

This document has not yet been approved by the
Government for general use or distribution

Table Of Contents

Tables and Figures

FIGURE 1. EVENT MESSAGE LIFE CYCLE 5

This document has not yet been approved by the

Government for general use or distribution

6.0 Overview

This document presents the functional requirements for the EVT within the Control Center System (CCS) of the Hubble Space Telescope (HST) under the control of Goddard Space Flight Center's (GSFC) Vision 2000 project . Requirements will detail the functionality of the EVT as well as interfaces of the EVT to other CCS subsystems.

This section defines events and clarifies the domain where they exist within HST operations. The definition and domain of corresponding event messages will also be defined. In addition, the event message life cycle will be detailed and illustrated.

6.0.1 Event Definition and Scope

Events are defined as actions or conditions within the scope of the HST, SN (including the ground network) , and CCS that are actively monitored for the following purposes:

- Optimizing testability in CCS development and integration.
- Aiding in health and safety analysis of the CCS, SN, and the HST during spacecraft operations.

Events are detected within the subsystems of the CCS; upon detection an event message is created within the corresponding subsystem.

6.0.2 Event Message Definition and Scope

Event messages are entities that are sent from one process to another within CCS to describe a particular occurrence of an event. All event messages contain basic information and functionality for the purpose of simplistic routing and processing. Some event messages may indicate where more detailed information may be found concerning the event being flagged.

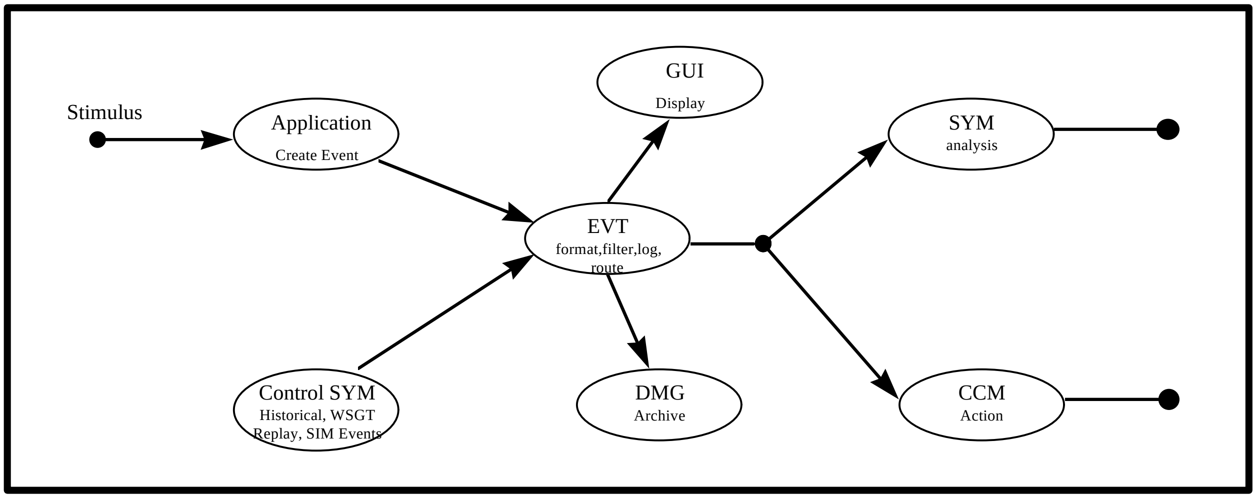
6.0.3 Event Message Life Cycle

The event message life cycle begins with a stimulus, or an event, that is detected within the various applications of the CCS. The event message is then generated and routed to the EVT where it is filtered, displayed, logged. All event messages are routed to a GUI server where client GUIs can display event messages.

Some filtered event messages are flagged and routed to various SYM processes for analysis. Other filtered event messages are filtered routed directly to the CCM for CCS infrastructure analysis. In addition, a CCS user may request that particular event messages be sent to a local process for analysis purposes. All filtered event messages are eventually killed by their destination process or killed by the EVT after being logged. Figure 1. illustrates the event message life cycle.

During Historical Modes historical event messages may be input into EVT from the Control SYM process. Historical event messages will be filtered, routed, but not logged.

Figure 1. Event Message Life Cycle



6.0.4 Managing Events Functionality

The EVT will support health and safety analysis during spacecraft operations and also aide in development and integration testing. The EVT will be the master control for all event messages created within the CCS. This will allow for enhanced maintainability and increased modularity for future extensions or modifications.

The EVT will perform the following functionality:

- Provide a mechanism for every application within the CCS to generate event messages. These event messages will then be routed to a central process for filtering, routing, and logging.
- Event messages will be routed to the DMG for long term storage.
- Filter and route event messages to other System Monitor (SYM) processes, the CCS Manager (CCM), GUI.

A CCS user may view event messages via a GUI that will display event messages in both graphical and textual formats. Event messages will be displayed graphically on a timeline according to the event message type and its source. The textual display of event messages will be displayed as ASCII text in a chronological log format. Displays will be color-coded for ease of interpretation. Event messages may be filtered according to the event message type, source, time, and textual searches.

> **Note:** Refer to GUI requirements for user interface descriptions. Refer to DMG requirements for logging and archiving descriptions. Refer to other SYM requirements for user generated event messages and historical event messages.

6.0.5 Examples

The following example illustrates how an event that occurs within the HST domain is processed by the CCS.

> The HST spacecraft battery-1 temperature reaches a yellow high limit. This event is detected via telemetry in the FEP and an event message is generated within the FEP. The

message is then routed to the EVT where it is filtered and routed to GUI. The EVT identifies this type of event message as one that needs to be routed to the SYM and GUI. The EVT routes the event message to the SYM and GUI.

The following example illustrates how an event that occurs within the SN domain is processed by the CCS.

The NCC TDRSS link is configured to state A. This event is detected via UPD and ODM streams in the FEP and an event message is generated within the FEP. The event message is then routed to the EVT where it is filtered and routed to GUI. The EVT identifies this type of event message as one that needs to be routed to the SYM. The EVT routes the event message to SYM.

The next example illustrates how an event that occurs within the CCS domain is processed by the CCS.

The ICS within the Command Processor receives a new command and is updated. This event occurs within the CMD subsystem and consequently the CMD generates an event message that is routed to the EVT where it is filtered and routed to GUI. The EVT identifies this type of event message as one that requires no further processing.

6.0 Manage Events Requirements

This section provides both the high level and detailed level requirements for the Manage Events process of CCS. Descriptions of the functionality provided by the EVT are provided.

6.1 High Level Event Processing

The EVT provides the Capability for CCS applications to generate and send event messages to the EVT process. The following subsystems will be allowed to generate and send event messages to the EVT:

- FEP
- Command Processing
- System Monitoring
- Data Management
- CCS Management
- GUI
- MDW

Note: Middleware (MDW) is not considered a CCS subsystem but will have the capabiltity to access the Event API for generating event messages and sending event messages to the EVT.

The FEP subsystem will generate event messages regarding the routing, display, and interpretation of downlink data concerning ODM/UPD and spacecraft telemetry. Uplink data concerning commands and table uploads will also be monitored within the FEP; corresponding event messages will be generated.

The CMD subsystem will generate event messages regarding command scheduling and processing. This includes receiving command requests, formatting, buffering, sending, and verifying spacecraft commands and Ground Control Message Requests (GCMR). Spacecraft orbital events such as ZOE and day/night transitions will also be generated by CMD.

The DMG subsystem will generate event messages regarding long term storage of CCS products. This includes items such as the Historical Integrated Command Schedule (HICS), telemetry, event log, event lookup table, and other products.

The CCM subsystem will generate event messages regarding CCS infrastructure and MUGSY configuration changes. This includes network status, HW status, SW status, time synchronization among CCS HW components, the IPC status among CCS components, and PDB changes.

The SYM system will generate event messages regarding the Control SYM, Distribute State Data, Detect Faults, Respond to Events & Faults, Perform Analysis and Trending, Perform Legacy, and Manage Monitor Data processes.

The GUI subsystem will generate event messages regarding the user interface software. This includes network and response message time outs, and user-related information such as login and logout activities.

6.0.1 The EVT shall provide the capability for every application within the CCS to generate event messages.

6.0.2 The EVT shall provide CCS developers with the capability to inject event messages into the system for testing purposes.

6.0.3 The EVT shall allow CCS applications to log event messages as text for testing purposes.

6.1 Detailed Level Event Processing

This process finishes building an event message by completing the type, severity, background, and a string version of the process name. This information is obtained from the DMG subsystem. After completing an event message, EVT will distribute the message to subscribing clients. The GUI subsystem will receive all messages.

To prevent logging multiple copies of event messages, any redundant real time EVT process running on the Core or Backbone LANs will not send event messages to the DMG. In addition, event messages generated in historical mode will not be logged.

Certain pre-determined event messages or event message types shall be flagged and routed to various SYM processes for analysis. These messages are typically concerned with HST or SN data that may affect the True State or Expected State data of the Fault Detection Expert System. An example of an event message type that may be filtered to the SYM are limit event messages that concern HST limit violations.

Certain pre-determined event messages or event message types shall be flagged and routed immediately to the CCM subsystem for further analysis and possible CCS configuration action. These messages are typically concerned with CCS infrastructure events that may affect CCS network, hardware, or software components. An example of an event message type that may be filtered to the CCM subsystem are error event messages that inform of the FEP shutdown.

6.1.1 Event messages shall have a subsystem source describing the CCS process that generated the event message.

Note: It is recommended that the following numbers identify the number ranges that are reserved for specific CCS subsystems.

- FEP
- CMD
- SYM
- DMG
- CCM
- GUI
- MDW

Note: Middleware (MDW) is not considered a CCS subsystem but will have the capability to access the Event API for generating event messages and sending event messages to the EVT.

6.1.2 Event messages shall have a node source describing the node of the application that generated the event message.

6.1.3 Event messages shall have a process name source describing the application that generated the event message.

6.1.4 Event messages shall have a process PID number describing the processes that generated the event message.

6.1.5 Event messages shall be categorized into one or more event types.

Note: Candidate Types include the following categories of event types:

- Telemetry Event
- Playback Event
- Real-time Event
- NCC Event
- Ground Configuration Event
- Alarm Event
- Command Dialog Event
- PSTOL Response Event
- General Event
- System or Software Error Event
- Keyboard Input Echo
- Command Page

6.1.6 Event messages shall have a time in UTC format to millisecond granularity describing the time that it was generated.

6.1.7 Event messages shall have unique identifier in integer format.

Note: The following lists the recommended number ranges that are reserved for specific subsystems

•	CCS	00,000 - 04,999
•	MDW	05,000 - 09,999
•	FEP	10,000 - 14,999
•	CMD	20,000 - 24,999
•	SYM	30,000 - 34,999
•	DMG	40,000 - 44,999
•	CCM	50,000 - 54,999
•	GUI	60,000 - 64,999

Note: The 1st and 2nd ID numbers for each range are reserved. The 1st number (e.g., 10,000) is reserved for a general (no Background data) message. The 2nd number (e.g., 10,001) is reserved for a general debug message (no Background data & severity = debug).

6.1.8 Event Messages shall have a status indicating the operational mode of the application that requested the event message.

6.1.9 Event messages shall have a textual description that details the specific information the message conveys.

6.1.10 Event messages shall have a severity level of 0, 1,2, or 3. Severity level 0 describes a debug event message. Severity level 1 describes a nominal operational without any urgency. Severity level 2 describes non-nominal situation with urgency. Severity level 3 describes an anomalous situation requiring immediate resolution.

6.1.11 The EVT shall have a status indicating if it is the single primary EVT process or a redundant Process

6.1.12 The EVT shall route all event messages to the DMG for archiving that meet the following criteria:

- From a real time application. The operational mode of the event messages is set to real time.
- The EVT processing this event message is a primary process.

Note: Middleware will provide a mechanism for guaranteeing delivery of event messages to the DMG.

6.1.13 The EVT shall complete all event messages that are not of historical operational mode.

6.1.14 The EVT shall flag event messages that need to be sent to the CCM or any process within the SYM for further analysis as they are received in the EVT.

6.1.15 The EVT shall route the flagged event messages to the appropriate subsystem or process.

6.1.16 The EVT shall route all event messages to the GUI subsystem.

Note: Guaranteed delivery of event messages to the GUI is not required.

List of Acronyms

API	Application Programming Interface
CCM	Control Center System Management
CCS	Control Center System
CDF	Common Data Format
CMD	Command Processor
CPU	Central Processing Unit
DMG	Data Management
EVT	Managing Events
FD	Fault Detection
FDF	Flight Dynamics Facility
FDIR	Fault Detection Isolation and Recovery
FEP	Front End Processor
FIR	Fault Isolation and Recovery
GCMR	Ground Control Message Request
GS	Ground System
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HST	Hubble Space Telescope
HW	Hardware
IPC	Inter-Process Communication
NCC	Network Control Center
ODM	Operational Data Message
P&S	Planning and Scheduling
PDT	Project Development Team
SN	Space Network
SW	Software
SYM	System Monitoring
TBD	To Be Determined
TDRS	Tracking and Data Relay Satellite
TDRSS	Tracking and Data Relay Satellite System
UPD	User Performance Data
UTC	Universal Time Clock
WS	White Sands

Requirements Source Matrix

Table 1 Requirements Source Matrix

Requirement Number	DLP Number	PORTS Requirement	ECS Requirement	Globalstar Requirement	Original Requirement
6.0.1					Original
6.0.2					Original
6.0.3			F-DMS-00920	GUI503.03, GUI505.02	
6.1.1		DSP901.5	F-DMS-01220	GUI501.03	
6.1.2					Original
6.1.3					Original
6.1.4					Original
6.1.5		DSP901.4	F-DMS-01220, F-DMS-01250, F-DMS-01260	GUI501.05	
6.1.6			F-DMS-01220	GUI501.02	
6.1.7		DSP901.2		GUI501.01	
6.1.8					Original
6.1.9		DSP904.1			
6.1.10			F-DMS-01250, F-DMS-01260	GUI504.01	
6.1.11					Original
6.1.12		DSP901.8	F-DMS-00910	GUI505.03	
6.1.13					Original
6.1.14				GUI502.03	
6.1.15				GUI502.03	
6.1.16				GUI502.03	